

8 – Sprites

Sprites diferentemente de animações possuem métodos que podem fazer a imagem se locomover pela tela.

A classe `Sprite` estende a classe `Animation`, isso quer dizer que tudo que fizemos na parte de animação também pode ser aplicado aos Sprites.

8.1 – Criando um Sprite

A classe `Sprite` possui dois construtores:

`public Sprite(String fileName, int numFrames)` - é passado o nome da imagem que será exibida e o número de frames que ela contém.

`public Sprite(String fileName)` - basta passar o nome da imagem a ser exibida, o número de frames automaticamente será 1.

Exemplo:

```
Sprite boneco = new Sprite("megaMan.png", 28);
```

Ou

```
Sprite bola = new Sprite("bola.png"); // o número de frames setados será 1
```

8.2 – Movendo o sprite

Para mover um sprite existem dois métodos:

`public void moveX()` – move o *Sprite* na tela somente no eixo x.

`public void moveY()` – move o *Sprite* na tela somente no eixo y.

Para determinar a velocidade com que o *Sprite* se moverá no eixo X ou no eixo Y pode ser os métodos

`public void setVelocityX(double velocity)`

`public void setVelocityY(double velocity)`

onde o parâmetro é a velocidade com a qual o *Sprite* deverá se deslocar no eixo x ou y.

`public double getVelocityX()` – retorna a velocidade de deslocamento do *Sprite* no eixo X.

`public double getVelocityY()` – retorna a velocidade de deslocamento do *Sprite* no eixo Y.

Obs.: as teclas padrões para fazer o *Sprite* se movimentar são as setas direcionais.

Para fazer com que o *Sprite* use outras teclas para se movimentar use os seguintes métodos:

`public void moveX(int leftKey, int rightKey)` – passe os códigos das teclas a serem usadas para locomover o *sprite* para a esquerda ou para a direita.

`public void moveY(int upKey, int downKey)` – passe os códigos das teclas a serem usadas para locomover o *sprite* para cima ou para baixo.

A chamada de algum dos métodos abaixo faz o *sprite* se locomover sem que seja preciso que alguma tecla seja pressionada:

`public void moveToUp()` – move o *sprite* para cima.

`public void moveToDown()` – move o *sprite* para baixo.

`public void moveToLeft()` – move o *sprite* para a esquerda.

`public void moveToRight()` – move o *sprite* para a direita.

Obs.: Todos os métodos apresentados não permitem que o *sprite* se locomova para fora da janela.

8.3 – Sentido de locomoção do *Sprite*

Para saber em qual sentido o *sprite* está se deslocando no eixo X use o método

```
public char getStateOfX()
```

que retorna se o *sprite* está se deslocando para a esquerda, para a direita ou se ele não está se locomovendo.

Para saber o sentido do deslocamento no eixo Y, use o método

```
public char getStateOfY()
```

que retorna se o *sprite* está se deslocando para cima, para baixo ou não está se deslocando.

Ambos os métodos retornam um *char*. Para determinar qual é o sentido, use as variáveis estáticas da classe *Sprite*: STOP, LEFT, RIGHT, UPWARD, DOWNWARD.

Exemplo:

```
sprite.moveX();
char sentidoX = sprite.getStateOfX()

if (sentidoX == Sprite.LEFT)
    System.out.println("esquerda");
else
    if (sentidoX == Sprite.RIGHT)
        System.out.println("direita");
    else
        System.out.println("parado");
```

Exemplo:

```
sprite.moveY();
char sentidoY = sprite.getStateOfY();

if (sentidoY == Sprite.DOWNWARD)
    System.out.println("descendo");
else
    if (sentidoY == Sprite.UPWARD)
        System.out.println("subindo");
    else
        System.out.println("parado");
```

Para saber se o *sprite* não está se movendo utilize: *Sprite.STOP*.

Exemplo 09: Determina o sentido de locomoção do *sprite* no eixo X e seta o frame correspondente a esse sentido.

```
public class Exemplo09
{
    public static void main(String[] args)
    {
        Window janela = new Window(800,600);
        Keyboard keyboard = janela.getKeyboard();

        GamelImage backGround = new GamelImage("paisagem.png");
        Sprite sprite = new Sprite("navio.png", 2);

        sprite.setPosition(300, 300);

        boolean executando = true;
```

```

while(executando)
{
    backGround.draw();
    sprite.draw();
    janela.display();

    sprite.moveX();

    if(sprite.getStateOfX()== Sprite.LEFT)
        sprite.setCurrFrame(0);
    else
        if(sprite.getStateOfX()== Sprite.RIGHT)
            sprite.setCurrFrame(1);

    if ( keyboard.keyDown(Keyboard.ESCAPE_KEY) == true)
        executando = false;
}
janela.exit();
}
}

```

8.4 – Fazendo o Sprite pular

Antes de fazer o *sprite* pular, devemos informar o valor da coordenada Y que servirá como chão, para isso use o método

```
public void setFloor(int floor).
```

Para fazer com que o *sprite* pule use o método **public void jump()**. Esse método usa a tecla SPACE para fazer o *sprite* pular.

Exemplo:

```

sprite.setFloor(500);
sprite.jump(); o sprite irá pular e ao cair não irá ultrapassar a coordenada y = 500;

```

Para controlar a velocidade e a altura do pulo use o método **public void setJumpVelocity(double velocity)** onde o parâmetro a ser passar é a velocidade inicial com que o *sprite* irá se deslocar do chão.

Para saber se o *sprite* ainda está executando o pulo use o método **public boolean isJumping()** que retorna *true* se o *sprite* não retornou ao chão, caso contrário, retorna *false*.

8.5 – Mudando a tecla usada no pulo

Por padrão a tecla a ser usada para fazer o boneco pular é a barra de espaço, caso você queira mudar essa tecla use o método

```
public void jump(int keyCode)
```

que terá como parâmetro o código da tecla que servirá para acionar o pulo.

8.6 – Simulando gravidade

Para simular o efeito de gravidade use o método **public void fall()**;

Para saber se o *sprite* tocou o chão use o método **public boolean isOnFloor()**, que retorna *true* se o chão já foi atingido pelo *sprite*, ao contrário, retorna *false*.

Assim como no pulo, o valor da coordenada Y que servirá como chão deve ser setada antes de usar o método `fall()`;

Para mudar o valor da gravidade use o método `public void setGravity(double gravity)`.

Exemplo 10: Simulando o efeito de gravidade

```
public class Exemplo10
{
    public static void main(String[] args)
    {
        Window janela = new Window(800,600);
        Keyboard keyboard = janela.getKeyboard();

        GameImage backGround = new GameImage("fundo.png");
        Sprite pedra = new Sprite("pedra.png".png");

        pedra.setPosition(300, 0);
        pedra.setFloor(500);
        pedra.setGravity(0.000098);

        boolean executando = true;
        while(executando)
        {
            backGround.draw();
            pedra.draw();
            janela.display();

            pedra.fall();

            if ( keyboard.keyDown(Keyboard.ESCAPE_KEY) == true)
                executando = false;
        }
        janela.exit();
    }
}
```

Exemplo 11: Troca a tecla usada para o pulo e faz o *sprite* pular

```
public class Exemplo11
{
    public static void main(String[] args)
    {
        Window janela = new Window(800,600);
        Keyboard keyboard = janela.getKeyboard();

        Keyboard.addKey(KeyEvent.VK_CONTROL);

        GameImage backGround = new GameImage("fundo.png");
        Sprite bola = new Sprite("baseball.png");

        bola.setPosition(300, 400);
        bola.setFloor(500);

        bola.setGravity(0.005);
        bola.setJumpVelocity(1);
        bola.setVelocityX(0.5);
    }
}
```

```

boolean executando = true;
while(executando)
{
    backGround.draw();
    bola.draw();
    janela.display();

    bola.moveX();
    bola.jump(KeyEvent.VK_CONTROL);

    if ( keyboard.keyDown(Keyboard.ESCAPE_KEY) == true)
        executando = false;
}
janela.exit();
}

```

8.7 – Movendo o sprite de um ponto a outro

Para mover o *sprite* de um ponto da tela até outro sem a intervenção do usuário use o método

```
public void moveTo(double x, double y)
```

Onde os valores passados como parâmetros são as coordenadas para onde se deseja que o *sprite* se desloque.